

Edimilson Batista dos Santos

**Reformulação de um Sistema de Tratamento e Reconhecimento de Padrões
para uma Microbalança de Quartzo utilizando Redes Neurais Artificiais**

Monografia de Graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências da disciplina Projeto Orientado para obtenção do título de Bacharel em Ciência da Computação.

Orientador
Prof. Joaquim Quinteiro Uchôa

Co-orientador
Prof. Mauro dos Santos de Carvalho

Lavras
Minas Gerais - Brasil
2002

Edimilson Batista dos Santos

**Reformulação de um Sistema de Tratamento e Reconhecimento de Padrões
para uma Microbalança de Quartzo utilizando Redes Neurais Artificiais**

Monografia de Graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências da disciplina Projeto Orientado para obtenção do título de Bacharel em Ciência da Computação.

Aprovada em 17 de dezembro de 2002

Prof. Mauro dos Santos de Carvalho (Co-orientador)

Prof. Joaquim Quinteiro Uchôa
(Orientador)

Lavras
Minas Gerais - Brasil

*Aos meus pais Geraldo e Maria
dos Anjos e irmãos Micaela e Roberto.
Com carinho especial, à minha namorada Valquíria.*

Agradecimentos

Agradeço a Deus por ter me ajudado a vencer mais uma etapa da minha vida.

A todos os professores que contribuíram com o meu sucesso.

Aos professores Joaquim Quinteiro Uchôa e Mauro dos Santos de Carvalho por terem me orientado no desenvolvimento deste trabalho.

E aos meus familiares e amigos, por terem me apoiado e incentivado durante os meus estudos.

Resumo

A classificação do café pela bebida é feita por "prova de xícara", sendo difícil e delicada, podendo sofrer influência pessoal do classificador. Uma alternativa para resolver esse problema é o uso de Redes Neurais Artificiais, como reconhecedores de padrões, para compor um nariz artificial capaz de avaliar os odores de café.

Este trabalho apresenta um sistema de tratamento e reconhecimento de odores de café, utilizando uma técnica de normalização para o pré-processamento dos dados de entrada e redes *Multi-Layer Perceptron(MLP)*, com topologias diferentes, treinadas com *backpropagation* padrão e com *backpropagation com momentum*.

Abstract

The classification of the coffee for the drink is made by "cup test", being difficult and delicate, therefore it suffers personal influence from the classifier. An alternative to decide this problem is the use of Artificial Neural Nets, as pattern recognition, to compose an artificial nose capable to evaluate the coffee odors.

This work presents a system of treatment and recognition of coffee odors, using one technique of normalization for the pay-processing of the entrance data and nets *Multi-Layer Perceptron(MLP)*, with different topologies, trained with *backpropagation* standard and *backpropagation with momentum*.

Sumário

1	Introdução	1
1.1	Uma aplicação das redes neurais	2
1.2	Classificação do café	3
1.3	Objetivos	3
2	O café	5
2.1	Importância do café	5
2.2	Classificação do café pela bebida	6
2.3	Alternativas para classificação da bebida	8
3	Narizes Artificiais	11
3.1	Histórico de Narizes Artificiais	11
3.2	Constituição de um Nariz Artificial	12
3.2.1	Sistema Sensor	12
3.2.2	Reconhecimento de Padrões	12
3.3	Aplicações de Narizes Artificiais	13
4	Redes Neurais	15
4.1	O Neurônio Biológico	15
4.2	Conceito de Redes Neurais	16
4.3	Desenvolvimento de Redes Neurais	19
4.3.1	<i>Perceptron</i>	20
4.3.2	Rede Neural <i>Backpropagation</i> <i>Backpropagation</i> Padrão	22
4.3.3	Rede Neural <i>Backpropagation</i> Variações	31
4.3.4	Outros modelos para redes neurais	35

4.4	Aplicações	35
5	Metodologia e Resultados Obtidos	37
5.1	Introdução	37
5.2	Descrição do Problema e da Base de Dados	38
5.3	Experimentos com <i>Multi-Layer Perceptron (MLP)</i>	39
5.3.1	Divisão do Conjunto de Padrões em Treinamento e Teste	39
5.3.2	Pré-processamento dos Dados	39
5.3.3	O Sistema de Reconhecimento de Padrões	40
5.3.4	Resultados Obtidos com MLP usando <i>Backpropagation</i> Padrão	42
5.3.5	Resultados obtidos com MLP usando <i>Backpropagation com</i> <i>Momentum</i>	47
6	Conclusões e Perspectivas Futuras	53
6.1	Conclusões	53
6.2	Perspectivas Futuras	54

Lista de Figuras

3.1	Esquema de um nariz artificial	12
4.1	Um neurônio natural.	15
4.2	Um modelo simples de neurônio artificial.	17
4.3	Uma rede neural muito simples.	18
4.4	Uma rede direta com 4 camadas de neurônios onde as conexões sinápticas são representadas apenas por setas.	18
4.5	Exemplos de como o modelo de neurônio pode calcular funções lógicas, com '1' representando Verdadeiro e '0' representando Falso. O número dentro das unidades representa seu limiar. Composto estas funções elementares, pode-se calcular qualquer função lógica.	20
4.6	O <i>perceptron</i> de Rosenblatt.	21
4.7	Um <i>perceptron</i> só pode distinguir entre classes linearmente separáveis, como as da esquerda. Em muitas funções isto não ocorre, como no caso da ou-exclusivo, à direita.	22
5.1	Um efeito da generalização comum em aprendizagem de redes neurais.	47

Lista de Tabelas

4.1	Valores para os parâmetros nos testes de Jacobs (1988).	34
4.2	Valores dos testes realizados por Jacobs (1988).	35
5.1	Exemplo de uma das repetições de aquisição de dados do café bebida dura.	38
5.2	Resultados obtidos para MLP com 4 unidades escondidas usando <i>backpropagation</i> padrão.	43
5.3	Resultados obtidos para MLP com 8 unidades escondidas usando <i>backpropagation</i> padrão.	44
5.4	Resultados obtidos para MLP com 20 unidades escondidas usando <i>backpropagation</i> padrão.	45
5.5	Resultados obtidos para MLP com 4 unidades escondidas usando <i>backpropagation com momentum</i>	48
5.6	Resultados obtidos para MLP com 8 unidades escondidas usando <i>backpropagation com momentum</i>	49
5.7	Resultados obtidos para MLP com 20 unidades escondidas usando <i>backpropagation com momentum</i>	50

Capítulo 1

Introdução

A capacidade de replicar um cérebro completamente funcional por meios puramente técnicos - excluindo, portanto, os métodos naturais e mágicos - equivaleria a possuir o entendimento final e completo do seu funcionamento. Ao longo da história da civilização humana, surgiram descrições mecanicistas do sistema nervoso, na tentativa de extrair sua essência funcional na forma de abstrações matemáticas que, em princípio, seriam reproduzíveis em alguma máquina.

Com a observação experimental do potencial de ação e de algumas topologias típicas, em que neurônios se interligam formando redes, surgem os modelos mecanicistas *bottom-up*, isto é, de baixo para cima, ou das partes em direção ao todo. O neurologista, filósofo e matemático Warren McCulloch (1988) é provavelmente um dos primeiros a gerar estes modelos, produzindo durante a sua carreira algumas dezenas de trabalhos seminais, reunidos mais tarde no livro *Embodiments of Mind* [McCulloch, 1988]. A linha originada por McCulloch é referida sob vários nomes como *Teoria Conexionista*, *Processamento Paralelo Distribuído* ou *Teoria de Redes Neurais* [Kovacs, 1997].

Segundo Kovacs (1997), as redes neurais recebiam atenção crescente. Desenvolvidas desde a década de 40 como possibilidade de um modelo de computação baseado no que se conhecia sobre o sistema nervoso, as redes neurais compunham-se de diversos processadores simples, cujo funcionamento era inspirado nos neurônios e que operavam em paralelo.

As redes neurais proporcionavam um método de representar relações completamente diferente de *máquinas de Turing* ou computadores com programas armazenados. Assim, com outros métodos numéricos, a disponibilidade de recursos computacionais, em *software* ou *hardware*, aumenta grandemente a utilidade da

abordagem, especialmente para grandes problemas.

1.1 Uma aplicação das redes neurais

Nos anos 80, as primeiras aplicações com redes neurais mais poderosas tornaram evidente que os elementos estatísticos com que a Inteligência Artificial (IA) simbólica se debatia há anos podiam, de um modo ou de outro, ser tratados por essas redes. Elas podiam, como aplicações típicas, reconhecer padrões, aprender com base em exemplos e tratar imprecisões e incertezas, realizando em certo grau aquilo em que a IA simbólica encontrava dificuldades.

Em meio ao entusiasmo com que se atacava os problemas que podiam ser resolvidos com as redes neurais, havia o sentimento entre alguns de que elas ampliavam os horizontes de todo o campo. No entanto, percebe-se, hoje em dia, que as aplicações de redes conexionistas permaneceram basicamente as mesmas: instâncias de reconhecimento de padrões, seja sobre sinais, imagens, sons ou uma série temporal. A grande maioria da pesquisa na área dedica-se ao aperfeiçoamento técnico, buscando melhorar o desempenho no reconhecimento de padrões [Braz, 1998].

Uma área a ser explorada nesse sentido é o reconhecimento de odores, sendo muito usado, por exemplo, no controle da poluição ambiental, nos diagnósticos médicos e nas indústrias de alimentos, de bebidas e cosméticos. Ainda hoje, esse processo é realizado por profissionais especializados, cujo olfato é treinado para os odores de interesse. Em geral, o custo deste treinamento é bastante alto e exige muito tempo até que sejam obtidos resultados satisfatórios. Além disso, a análise de odor pelo ser humano é subjetiva e está sempre sujeita a variações provocadas pelo cansaço, por doenças e até pelo estado emocional do indivíduo.

Sendo assim, fica evidente a necessidade de desenvolver narizes artificiais, que são sistemas capazes de detectar e classificar odores, vapores e gases automaticamente. Para reconhecer os padrões gerados pelos elementos sensores quando em contato com odorantes, têm sido amplamente utilizadas as redes neurais artificiais, que apresentam diversas vantagens, como a capacidade de lidar com sinais não-lineares provenientes dos sensores, a adaptabilidade, a tolerância a erros, a tolerância ao ruído e o paralelismo inerente, permitindo rapidez no uso após o treinamento. A rede mais comumente utilizada é o *Perceptron Multi-Camadas ou Multi-Layer Perceptron (MLP)*, que é treinado com o algoritmo *Backpropagation*.

Os narizes artificiais trazem vantagens para diversas áreas. Por exemplo, em relação ao monitoramento ambiental, existe a necessidade de analisar misturas

combustíveis, detectar vazamentos e controlar a qualidade do ar e as emissões industriais. Os diagnósticos médicos podem ser facilitados, uma vez que os odores da respiração representam sintomas de problemas gastrointestinais, infecções, diabetes e doenças do fígado. Além disso, tecidos infectados emitem odores característicos. Já na indústria de alimentos, os narizes artificiais tem aplicações no controle da qualidade e no monitoramento de processos de produção. Na cafeicultura, uma área de grande interesse comercial, podem ser usados na classificação do café, como uma ferramenta de auxílio à "prova de xícara".

1.2 Classificação do café

Sendo o café negociado comercialmente na forma de "café em grão" ou "café verde", isto é, depois de colhido, preparado, secado, beneficiado e ensacado, há uma classificação comercial que determina a qualidade do produto de acordo com seis itens principais: cor, peneira, tipo, torração, bebida e descrição ou características.

A bebida, conseqüência final de todos esses itens, é determinada pela degustação da bebida "café", depois de feita a infusão. Essa classificação de café é feita por "prova de xícara", cuidadosa e técnica, e pode distinguir as seguintes classes principais: dura, mole e rio. Dessas três classes principais ainda surgem as intermediárias.

Entretanto, a classificação pela bebida é difícil e delicada, sofrendo influência pessoal do classificador. A subjetividade da prova de xícara é bastante discutida e, embora a determinação da qualidade da bebida seja passível a erros devido à discrepância do paladar, torna-se difícil encontrar outra solução tendo em vista a complexidade dos vários fatores que a afetam.

1.3 Objetivos

O objetivo deste trabalho é propor uma reformulação de um sistema de tratamento e reconhecimento de padrões de odores de café utilizando redes neurais artificiais, treinadas com o algoritmo de aprendizagem *backpropagation* e *backpropagation com momentum*. O sistema original foi implementado por Oliveira (2002) e despertou o interesse pela concretização de uma ferramenta de auxílio à prova de xícara que pudesse facilitar a classificação do café.

Outro objetivo, é a apresentação de um estudo sobre a importância do café e

sua classificação, destacando algumas alternativas desenvolvidas para essa classificação.

Além disso, o trabalho apresenta um breve histórico sobre o desenvolvimento dos narizes artificiais, e, em seguida, são analisadas suas partes constituintes. Também, apresenta um resumo sobre as redes neurais, enfatizando o algoritmo de aprendizagem (ou treinamento) *backpropagation* e as variações existentes que melhoram o seu desempenho.

Capítulo 2

O café

2.1 Importância do café

Dá-se o nome de "café" ao fruto ou à semente do cafeeiro; a palavra também é usada para designar a infusão feita com a semente torrada e moída ou o estabelecimento de preparo e venda da bebida diretamente ao público. Alguns autores, segundo Graner (1967), admitem que a palavra "café" tenha se originado de "*Cafa*", região de sua origem na Etiópia, África. O produto econômico é uma bebida da cor da pele do povo desse país.

Dotada de sabor e de perfume característicos, vem, através do tempo, arrasando sua história e a do povo que disputou o seu uso. A fama dessa bebida, considerada como poderoso e inofensivo excitante do sistema nervoso, especialmente do cérebro, deu-lhe renome suficiente para transpor, através de mercadores e de navegantes, montanhas e mares, disseminando-se por todo o mundo.

Como bebida, obtida, seja das folhas ou dos frutos maduros, foi também conhecida da medicina desde muito tempo, não só na Arábia como também na Europa e nas Américas. Foi tida como: estimulante do cérebro; confortadora para dores de cabeça e para tosse; cura para desmaios e para raquitismo; útil como moderadora de beberrões; aconselhada para aliviar os incômodos periódicos das mulheres árabes; boa para reumatismo, febres intermitentes e para combater vermes nas crianças. Atualmente, é recomendada como ótimo estimulante.

Várias dificuldades surgiram ao longo dos tempos para que, na primeira metade do século XVII, o café estivesse sendo consumido nas altas rodas da sociedade londrina e, logo mais tarde, nas classes menos abastadas. Na Inglaterra, houve taxaço de pesado imposto sobre os estabelecimentos. Muitos interessa-

dos no fechamento dos cafés públicos apareceram, como fabricantes de cerveja e aguardente, os quais tiveram até o apoio das mulheres. Mas, o prestígio da bebida foi tanto, que se instalaram muitos desses cafés públicos pela cidade de Londres.

Na Prússia, não conseguindo evitar a preferência do povo, Frederico, o Grande, resolveu estabelecer um monopólio da exploração do café pelo governo e tirar proveito dessa preferência [Graner, 1967].

Hoje, o café continua como uma das bebidas mais consumidas no mundo e sua comercialização é caracterizada por constituir-se de produtos primários sujeitos a grandes flutuações de preços decorrentes dos desequilíbrios entre a oferta e a demanda. Estas freqüentes flutuações fizeram com que países produtores e consumidores se esforçassem conjuntamente na formalização de Acordos Internacionais de "*commodities*".

O principal objetivo dos Acordos Internacionais de "*commodities*" é a estabilização de preço, operando através de mecanismos reguladores, tais como cotas de exportação, faixa de preços máximo e mínimo e estoques reguladores.

De acordo com Junior (1995), o café é uma das mais importantes "*commodities*" comercializadas no mercado internacional, constituindo-se numa das principais fontes de divisas que propiciam o crescimento e desenvolvimento das economias menos desenvolvidas.

2.2 Classificação do café pela bebida

Segundo Graner (1967), a classificação comercial do café é feita segundo seis itens principais: cor peneira, tipo, torração, bebida e descrição ou características. Feita a classificação da torração, que é anotada no respectivo cartão de classificação, preso à bandeja de alumínio, devemos passar à prova de xícara, ou seja, a degustação da bebida do café.

A bandeja de alumínio, que se encontra sobre a mesa de prova, está acompanhada do cartão de classificação, no qual já se encontram consignadas as demais classificações, com exceção da bebida, ou sejam: cor, peneira, tipo, torração, descrição ou características.

A mesa de prova consta de uma superfície circular, giratória, de granito ou material semelhante, sustentada por um pé central ao qual está preso um braço não giratório, contendo uma pequena bandeja, onde ficam duas xícaras de mais ou menos 150cm^3 , onde se coloca água quente ou fria. No seu bordo, há uma depressão em forma de coroa, onde ficam as xícaras, também de 150cm^3 , sendo 3 (ou mais) em frente de cada bandeja.

Com a colher de prova em forma de concha pequena, semelhante a uma colher de açucareiro, com o cabo um pouco maior, retiram-se aproximadamente 10g de pó, que se colocam em cada xícara. É importante que sejam colocadas, em cada xícara, sempre quantidades iguais de pó, para permitir ao "classificador-provador", não só testar a qualidade da bebida, como também a sua fortidão e encorpamento.

Enchidas as xícaras com água fervida (não fervida por muito tempo), inclusive as da bandeja do braço não giratório, vai-se mexendo com a própria colher de prova, xícara por xícara; ao passar de uma amostra para a outra, isto é, de cada 3 xícaras de uma amostra para as 3 xícaras da outra amostra, deve-se lavar a colher em água fervente, que deverá estar disponível no braço não giratório.

Depois de preparadas as xícaras, o classificador cheira a infusão feita para ter uma idéia prévia da qualidade do café. Em seguida, a mesa recebe um pequeno impulso, começando a girar para o esfriamento do café e o assentamento do pó no fundo das xícaras. Tendo a infusão adquirido uma temperatura que a torne suportável ao paladar, começa-se a "prova de xícara".

O classificador senta-se ao lado da mesa de prova, tendo à sua direita a bandeja do braço não giratório, contendo duas xícaras com água morna e limpa para lavar a colher e a boca; a sua direita também, mais ao lado, fica localizada uma cuspidreira de metal; a seguir, toma a colher de prova, colocada em uma dessas xícaras, na bandeja, enche-a de café e leva-a à boca; depois, com um chupão bem forte, faz com que a infusão acompanhada de ar, penetre forte e profundamente dentro da boca. Com movimentos rápidos da língua e da bochecha, o classificador faz com que a infusão vá repetidamente de encontro à língua e ao palatino, depois a cospe na cuspidreira.

Depois de provar duas ou três vezes a mesa toda, o classificador anota as classificações nos cartões, e dá o lugar para o outro classificador-provador, que virá então confirmar as suas determinações ou anotar qualquer divergência. Havendo divergências, prevalece a opinião da maioria ou a do classificador-chefe, que possui em geral maior prática e mais conhecimentos técnicos.

A classificação pela bebida é difícil e delicada, estando exposta à subjetividade do classificador. Também as variações do paladar, em virtude de diferentes alimentos ingeridos no dia, ou no dia anterior, o grande número de classificações feitas numa mesma ocasião podem influir nos resultados da classificação. Assim, diante dessas dificuldades, o classificador precisa cercar-se de todo cuidado controlando e conferindo o seu trabalho, para evitar cair em erros inevitáveis nessas circunstâncias. A subjetividade da prova da xícara é bastante discutida e, embora a determinação da qualidade da bebida seja passível a erros devido à discrepância

do paladar, torna-se difícil encontrar outra solução tendo em vista a complexidade dos vários fatores que a afetam.

Geralmente, um bom classificador-provador, bem treinado, pode provar cerca de 10 mesas por dia, com 12 amostras por mesa, num total de 120 amostras ou 360 xícaras. No entanto, há classificadores que chegam a provar durante o dia todo, sem grandes alterações no seu paladar, apresentando, todavia, falhas na precisão e na segurança das suas classificações. Geralmente, essas classificações em massa são feitas com o fito de seleções, que para venda quer para compra e, nessas circunstâncias, esses cafés serão novamente classificados, ou seja, dependerão de uma reclassificação.

A classificação de café, por "prova de xícara" cuidadosa e técnica, pode distinguir as seguintes classes principais:

- Dura: quando apresenta gosto acentuadamente áspero, desagradável e adstringente.
- Mole: quando o gosto é suave, adocicado, ou com acidez suave, agradável ao paladar.
- Rio: quando apresenta gosto forte, característico, lembrando iodo-fórmio ou ácido fênico.

Dessas três classes principais ainda surgem as intermediárias, as quais são facilmente distinguidas porque preponderantemente fazem parte da gama maravilhosa e infinita do paladar dos cafés. Destas, as que são mais usadas em classificação de café, são as seguintes: "Apenas mole" entre a dura e a mole, "Estritamente mole" melhor do que a mole, "Rio" que se divide em 4 modalidades, a saber: "Levemente riada" ou "Dura com fundo Rio"; "Riada"; "Rio" e "Rio Macaco" ou "Rio zona da mata".

Segundo Graner (1967), as designações mais usadas na classificação da bebida do café, na ordem da melhor para pior, são as seguintes: Estritamente mole, Mole, Apenas mole, Dura, Levemente Riada ou Dura com fundo Rio, Riada, Rio, Rio Macaco ou Rio Zona da Mata.

2.3 Alternativas para classificação da bebida

De acordo com Pimenta (1995), na tentativa de encontrar outra solução para a classificação do café, têm sido realizados trabalhos exaustivos, visando correlacionar

a composição química, atividade de polifenoloxidasas e peroxidases do grão com a qualidade de bebida.

O teor de acidez titulável em grãos de café pode variar de acordo com os níveis de fermentações que ocorrem nos grãos e também com os diferentes estádios de maturação dos mesmos, podendo também servir como suporte para auxiliar na avaliação da qualidade de bebida do café.

Carvalho *et al.* (1994) verificaram haver diferenças marcantes nos valores de acidez titulável em cafés com diferentes qualidades de bebida, ou seja, valores médios de 211,2; 235,5; 218,3; 250,4; 272,2 e 284,5 ml NaOH/100g de amostra respectivamente para café de bebida "estritamente mole", "mole", "apenas mole", "dura", "riada" e "rio". Ressaltaram a importância da utilização desta acidez, junto à atividade da polifenoloxidase e índice de coloração como suporte para uma maior eficiência da classificação sensorial.

As enzimas polifenoloxidasas atuam sob os compostos fenólicos e se encontram ligadas as membranas celulares sendo ativadas somente quando liberadas destas, e de acordo com vários autores, se mostram diretamente relacionadas com a qualidade de bebida do café. Amorim (1978) descreve em seus trabalhos "*in vivo*" que a enzima polifenoloxidase tem sido encontrada na polpa de frutos e nas camadas externas e partes centrais do grão.

Carvalho *et al.* (1994) verificaram haver variações da atividade da polifenoloxidase, que permitem separar as classes de bebida com base nas atividades destas enzimas, constatando assim um aumento significativo na atividade da polifenoloxidase à medida que o café apresenta-se de melhor qualidade.

Além das pesquisas relacionadas com a composição química do café, sistemas analíticos específicos, ou narizes artificiais, estão sendo desenvolvidos para o reconhecimento de odores de café.

Oliveira (2002) propôs a construção de um sistema de reconhecimento de odores de café, utilizando *redes neurais backpropagation*, para compor um nariz artificial capaz de auxiliar na classificação do café. Os resultados dos experimentos foram promissores, mas incompletos, pois os padrões, obtidos por um dispositivo sensor do tipo microbalança de quartzo, foram insuficientes para treinar as redes neurais da maneira como foram implementadas.

Capítulo 3

Narizes Artificiais

Uma grande variedade de gases e vapores precisam ser analisados em diversas aplicações, como controle de qualidade em indústrias de alimentos, bebidas e cosméticos, monitoramento ambiental e diagnóstico médico. Muitas destas análises são feitas pelo olfato de indivíduos especializados, o que apresenta uma série de fatores desfavoráveis, como a influência de alergias e doenças, os riscos da inalação de certas substâncias tóxicas, a fadiga e a mudança da sensibilidade olfativa de acordo com o estado mental.

Segundo Yamasaki (2001), a necessidade de sistemas analíticos específicos para o reconhecimento de odores vem se tornando cada vez mais evidente. Várias áreas demandam sistemas desse tipo. Este sistema analítico para o reconhecimento de odores é denominado muitas vezes nariz artificial ou eletrônico.

3.1 Histórico de Narizes Artificiais

De acordo com Yamasaki (2001), provavelmente, o primeiro equipamento criado com a finalidade de detectar odores foi desenvolvido por Moncrieff em 1961 [Moncrieff, 1961]. Tal sistema era mecânico, sendo que os primeiros narizes eletrônicos foram desenvolvidos por Wilkens e Hatman em 1964 [Wilk&Hat, 1964], Buck *et al.* em 1965 [Buck et al., 1965] e Dravnieks e Trotter, também em 1965 [Dravnieks e Trotter, 1965]. Entretanto, o conceito de nariz artificial como um sistema inteligente com sensores químicos para a classificação de odores só apareceu em 1982, nos trabalhos de Persaud e Dodd [Persaud e Dodd, 1982], e em 1985, nos trabalhos de Ikegami e Kaneyasu [Ikegami e Kaneyasu, 1985].

3.2 Constituição de um Nariz Artificial

Um nariz artificial é formado por duas partes principais: um *sistema sensor* e um *sistema de reconhecimento de padrões*. A estrutura básica de processamento de dados em um nariz artificial pode ser ilustrada pela figura 3.1.

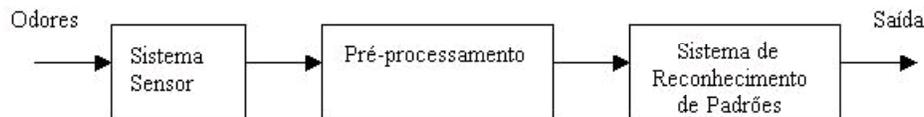


Figura 3.1: Esquema de um nariz artificial

3.2.1 Sistema Sensor

O sistema sensor pode ser composto por um conjunto de sensores distintos, em que cada elemento mede uma propriedade diferente do composto odorante, ou por um único dispositivo (por exemplo, um espectrômetro), que produz um conjunto de medições para cada odorante. Pode ser constituído, ainda, por uma combinação de ambas as estruturas [Keller et al., 1995].

Cada composto odorante apresentado ao sistema sensor produz um padrão característico. É importante examinar cuidadosamente os dados gerados pelo conjunto de sensores. Tais dados devem ser preparados da melhor forma possível. Normalmente, é necessária uma etapa de pré-processamento dos sinais gerados pelo sistema sensor antes da fase do reconhecimento de padrões. Em seguida, apresenta-se este sinal ao sistema de reconhecimento de padrões.

3.2.2 Reconhecimento de Padrões

Em geral, os sistemas de reconhecimento de padrões são divididos em paramétricos e não-paramétricos [Gardner e Hines, 1997]. Uma técnica paramétrica assume que os dados dos sensores podem ser descritos por uma função densidade de probabilidade. Normalmente, admite-se que os dados têm uma distribuição normal, com média e variância conhecidas. Este tipo de técnica exige a construção de uma base de conhecimento, contendo as funções densidade de probabilidade apropriadas. As técnicas de reconhecimento de padrões que não assumem funções densidade de probabilidade são conhecidas como não-paramétricas.

As principais técnicas de reconhecimento de padrões utilizadas em narizes artificiais são: análise de função discriminante, análise de clusters, análise das componentes principais, redes neurais artificiais [Gardner e Hines, 1997]. Mais detalhes sobre essas técnicas podem ser encontrados em Yamasaki (2001).

Redes neurais artificiais são sistemas não-paramétricos e têm apresentado diversas vantagens no reconhecimento de padrões de odor, como a capacidade de lidar com sinais não-lineares provenientes dos sensores, adaptabilidade, tolerância a erros, tolerância a ruído e paralelismo inerente, permitindo rapidez no uso após o treinamento. O tipo mais popular é o *Multi-Layer Perceptron* (MLP), que é treinado com o algoritmo *backpropagation*. Maiores detalhes sobre redes MLP e algoritmo *backpropagation* podem ser encontrados no capítulo 4 do presente trabalho.

Exemplos de utilização de redes neurais para narizes eletrônicos podem ser vistos em Yamazaki (2001).

3.3 Aplicações de Narizes Artificiais

Apesar de já existir muitos narizes artificiais disponíveis no mercado, o nariz humano ainda é considerado o mais importante instrumento analítico para garantia da qualidade do odor em muitas indústrias de bebidas, alimentos e perfumaria. Entretanto, o interesse por melhoria na sua tecnologia e investimento em pesquisa ainda é alto em razão do grande número de aplicações de interesse comercial que podem ser desenvolvidas. Por exemplo, treinar pessoas para desenvolver estas atividades requer um alto custo, em razão dos especialistas só poderem trabalhar em curtos períodos de tempo, e os equipamentos utilizados para desenvolver estas atividades, não só consomem tempo, como também são muitas vezes inadequados. Portanto, um grande número de aplicações tem surgido para análise ambiental, indústria de alimentos, bebidas, medicina, perfumaria, robótica e segurança [Oliveira, 2002].

O nariz artificial pode ser útil para realizar o controle ambiental uma vez que ele pode ser utilizado para fazer a identificação em tempo real no campo de resíduos tóxicos, análises de misturas de combustível, detecção de vazamento de óleo, detecção de gás, identificação de odores domésticos, monitoramento da qualidade do ar e poluentes de água.

Outra aplicação do nariz na indústria pode ser feito no controle de alimentos e bebidas que abrange desde qualidade de alimentos, monitoramento de processos de fermentação, inspeção de bebidas, classificação de *whisky* até condições de estocagem.

Capítulo 4

Redes Neurais

4.1 O Neurônio Biológico

Para entender melhor o que são as Redes Neurais Artificiais (RNAs), é necessário conhecer a estrutura do sistema nervoso. A seguir serão apresentadas as principais características de um neurônio biológico.

O cérebro é constituído por aproximadamente 10 bilhões de células nervosas (neurônios). Cada célula possui três componentes: pequenas ramificações denominadas dendritos, o corpo celular (também conhecido como soma) e o axônio (figura 4.1). A conexão entre o axônio de um neurônio e o dendrito de outro é chamada de sinapse. As células nervosas estão interconectadas através das sinapses. O axônio de um neurônio pode estar conectado a dendritos de vários outros neurônios.

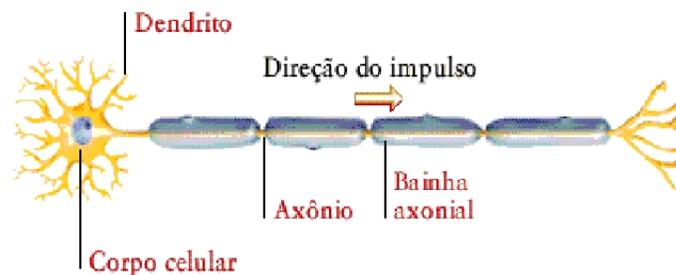


Figura 4.1: Um neurônio natural.

Os estímulos do meio externo são captados pelos órgãos sensores e conduzidos até o cérebro, onde são recebidos pelos dendritos dos neurônios. Em seguida, ocorrem reações químicas e elétricas no soma, ou seja, o processamento da informação. A seguir, a saída do soma, em forma de impulsos elétricos inibidores ou excitadores, é propagada para os neurônios vizinhos através das sinapses.

4.2 Conceito de Redes Neurais

Segundo Laurene Fausett, *uma rede neural artificial (RNA) é um sistema de processamento de informação que tem certamente características de desempenho em comum com redes neurais biológicas* [Fausett, 1994]. Redes neurais artificiais têm sido desenvolvidas como generalizações de modelos matemáticos de cognição humana ou biologia neural, baseadas nas suposições de que:

1. o processamento de informação ocorre em elementos simples chamados neurônios;
2. sinais são passados entre neurônios através de conexões;
3. cada conexão tem um peso associado, o qual, em uma rede neural típica, multiplica o sinal transmitido;
4. cada neurônio aplica uma função de ativação (usualmente não linear) a sua entrada para determinar seu sinal de saída.

Uma rede neural é caracterizada por:

1. seu padrão de conexões entre os neurônios (conhecido como sua arquitetura);
2. seu método de determinar os pesos nas conexões (conhecido como seu *treinamento*, ou *aprendizagem*);
3. sua função de ativação.

A rede neural consiste de um grande número de elementos de processamento chamados neurônios, unidades, células ou nós. Cada neurônio é conectado com outros neurônios de forma a se comunicar diretamente, cada um com um peso associado. O peso representa informações sendo usadas pela rede para resolver um

problema. Redes neurais podem ser aplicadas a uma grande variedade de problemas, como armazenar e recuperar dados ou padrões, classificar padrões, executar mapeamentos generalizados de padrões de entrada para padrões de saída, agrupar padrões similares, ou encontrar soluções para problemas de otimização.

Cada neurônio tem seu estado interno, chamado de sua ativação ou nível de atividade, o qual é uma função de entrada. Tipicamente um neurônio envia sua ativação como um sinal para vários neurônios. É importante notar que um neurônio pode enviar apenas um sinal por vez, apesar do sinal ser espalhado para vários neurônios.

Por exemplo, considere um neurônio Y (figura 4.2) que recebe entrada de neurônios X_1 , X_2 e X_3 . As ativações (sinais de saída) desses neurônios são x_1 , x_2 e x_3 , respectivamente. Os pesos nas conexões de X_1 , X_2 e X_3 para Y são w_1 , w_2 e w_3 , respectivamente. A entrada da rede, y_{in} , para o neurônio Y , é a soma dos pesos dos neurônios X_1 , X_2 e X_3 , ou seja,

$$y_{in} = w_1x_1 + w_2x_2 + w_3x_3$$

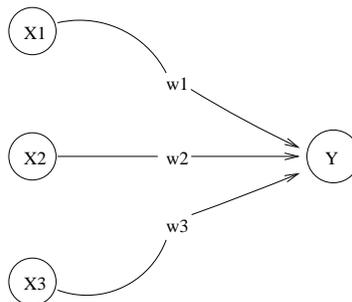


Figura 4.2: Um modelo simples de neurônio artificial.

A ativação y do neurônio Y é dada por alguma função de sua entrada, $y = f(y_{in})$, por exemplo, uma função logística binária (uma curva em forma de S):

$$f(x) = \frac{1}{1 + \exp(-x)}$$

Agora suponha que o neurônio Y é conectado aos neurônios Z_1 e Z_2 , com pesos v_1 e v_2 , respectivamente (figura 4.3). O neurônio Y envia seu sinal y para cada um. Mas, em geral, o valor recebido pelo neurônio Z_1 e Z_2 seriam diferentes, porque cada sinal é acompanhado pelo peso apropriado v_1 e v_2 . Em uma rede comum, as ativações z_1 e z_2 dos neurônios Z_1 e Z_2 seria dependentes das entradas

de vários neurônios, não apenas um como mostrado nesse exemplo.

Apesar da rede neural da figura 4.3 ser muito simples, a presença de uma unidade oculta, junto com uma função de ativação não linear, descreve a sua habilidade para resolver muito mais problemas do que numa rede com apenas unidades de entrada e saída. Por outro lado, é muito mais difícil de ser treinada (encontrar valores ótimos para os pesos).

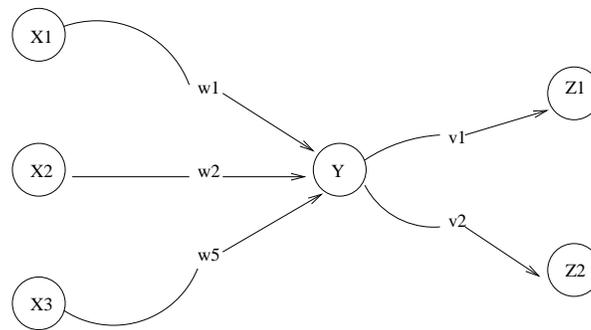


Figura 4.3: Uma rede neural muito simples.

A figura 4.4 apresenta outro exemplo de rede neural. Entre tantas topologias existentes, ela destaca uma topologia particular de rede neural, as Redes Diretas. Em princípio, as topologias de RNAs são representadas por grafos dirigidos como esse.

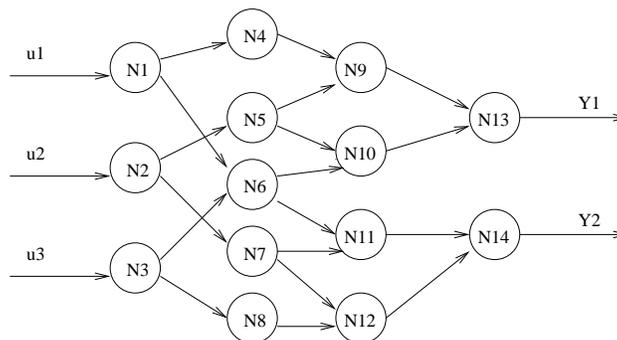


Figura 4.4: Uma rede direta com 4 camadas de neurônios onde as conexões sinápticas são representadas apenas por setas.

Redes Diretas (*Feedforward*) são aquelas cujo grafo não tem ciclos. É comum representar estas redes em camadas e neste caso são chamadas *redes em camadas*. Neurônios que recebem sinais de excitação são chamados de *camada de entrada*, ou *primeira camada*. Neurônios que tem sua saída como saída da rede, pertencem à *camada de saída* ou *última camada*. Neurônios que não pertencem nem a camada de entrada, nem a camada de saída, são neurônios internos à rede e pertencem a uma ou mais *camadas internas ou ocultas (hidden layers)*.

Para que haja sentido falar em redes por camadas, é necessário que os neurônios internos se organizem formando camadas. A camada de entrada é assim chamada por receber todas as entradas do meio exterior e saída nos outros neurônios e de forma análoga à camada de saída. Se os neurônios internos se organizam em grupos, cada um recebendo entradas somente de um grupo anterior (entrada, etc) e suas saídas vão para outro grupo, tem-se uma rede em camadas.

Essas redes são atualmente as mais populares, principalmente por existirem métodos de aprendizado fáceis de usar. Um método muito usado é o *backpropagation*, que denominou esse tipo de rede de *Rede Backpropagation ou redes de retropropagação*. Além disso, essas redes são capazes de aproximar, com maior ou menor precisão, dependendo do número de neurônios da rede, qualquer função não linear. Entretanto, mesmo no caso de usarem neurônios dinâmicos (equação diferencial de primeira ordem ou a uma diferença finita), tem uma dinâmica muito limitada não podendo representar todos os sistemas dinâmicos [Barreto, 2001].

4.3 Desenvolvimento de Redes Neurais

McCulloch e Pitts, a partir de seu modelo de neurônio (figura 4.2), mostraram que qualquer função lógica podia ser implementada, através da composição das funções lógicas básicas 'e', 'ou' e 'não', implementadas pelo modelo da figura 4.5 [McCulloch & Pitts, 1943].

Então, surgiu interesse em modelos de redes neurais que pudessem reproduzir procedimentos mentais, através da compreensão detalhada da circuitaria cerebral. Mas, embora lançassem uma ponte entre a neurologia e a computação, não era claro como utilizá-las para atingir este objetivo. Um dos grandes obstáculos, por exemplo, era reproduzir o aprendizado, entendido como a adaptação crescente ao ambiente através da experiência. Isso permitiria que redes fossem construídas apenas com a interação automática com o ambiente ou uma simulação deste, uma natural e atraente alternativa à montagem exaustiva de redes que realizassem tarefas inteligentes.

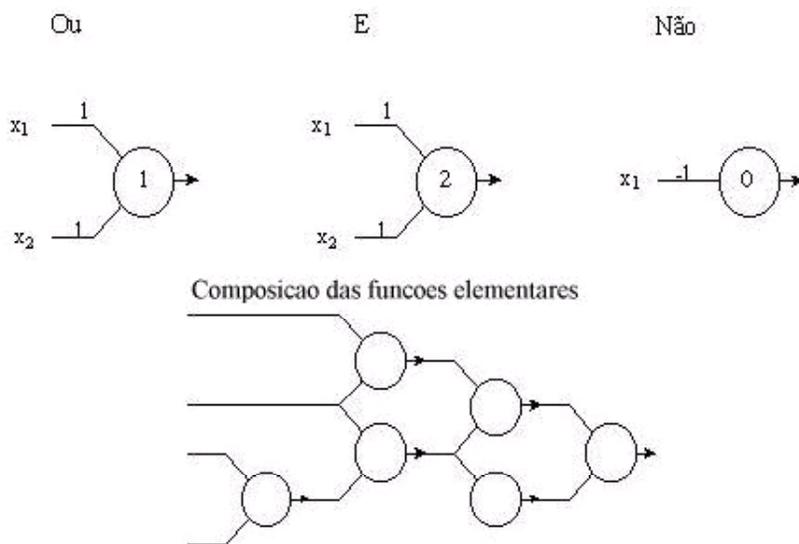


Figura 4.5: Exemplos de como o modelo de neurônio pode calcular funções lógicas, com '1' representando Verdadeiro e '0' representando Falso. O número dentro das unidades representa seu limiar. Compondo estas funções elementares, pode-se calcular qualquer função lógica.

4.3.1 Perceptron

Donald Hebb (1949) fez algumas especulações em seu livro *Organizations of Behaviour* de que o conhecimento no sistema nervoso se baseava nas conexões entre os neurônios e que, portanto, o aprendizado devia se refletir na força destas conexões [Hebb, 1949]. Apesar disso, apenas em 1958 surgiu um primeiro modelo de rede conexionista artificial que podia aprender. Desenvolvido por Frank Rosenblatt, o chamado *Perceptron* chamou a atenção por suas características computacionais peculiares [Rosenblatt, 1958]. Voltado principalmente para a reprodução do processamento visual no sistema nervoso, ele consistia de um conjunto de unidades, ou neurônios artificiais, organizados em três camadas: uma de entrada (ou retina), uma intermediária (ou de predicados) e uma de saída (ou de decisão) (figura 4.6). Quando uma entrada era apresentada a essa máquina, ela era representada na retina através de ativações nas unidades dessa camada. Cada unidade da camada de predicados, então, através de suas conexões com um subconjunto da retina, podia ou não disparar, ou tornar-se ativada, de acordo com os pesos das conexões. As unidades de decisão, finalmente, usavam esses predicados para cal-

cular a função final, através dos pesos de suas conexões com elas. Esses pesos das conexões, entre a camada de predicados e a de decisão, eram modificados de acordo com a decisão correta desejada, fornecida por um supervisor.

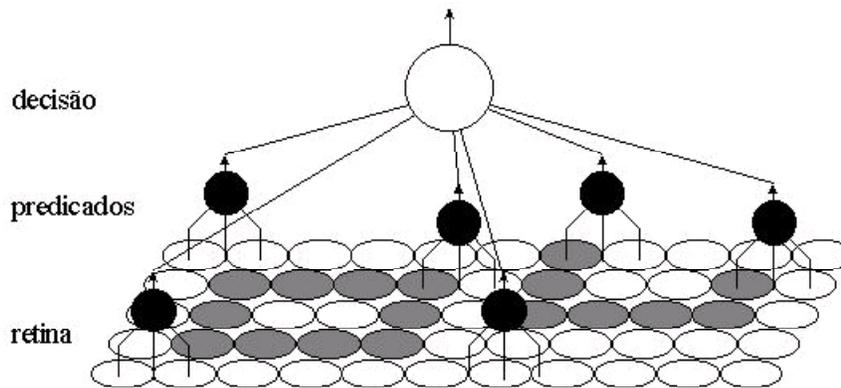


Figura 4.6: O *perceptron* de Rosenblatt.

O resultado era uma máquina que podia aprender a classificar um conjunto arbitrário de padrões, a partir de exemplos. Os pesos das conexões alteravam-se sucessivamente, adaptando-se às respostas desejadas e tornando a máquina cada vez mais correta.

Rosenblatt (1958) provou um importante teorema mostrando que qualquer solução representável por um *perceptron* podia ser aprendida por ele. Porém, havia sérias limitações sobre o que podia ser aprendido por ele.

Em um importante trabalho, *Perceptrons*, Minsky e Papert (1969) exibiram de maneira rigorosa as diversas limitações computacionais dos modelos até ali existentes [Min&Papert, 1969]. Uma delas consistia no fato de que algumas funções muito simples, como a função booleana ou-exclusivo, não podiam ser aprendidas por um *perceptron*. Isto acontece porque um *perceptron* é, fundamentalmente, um discriminador linear, ou seja, pode apenas distinguir, no espaço de suas entradas, entre as duas classes pertencentes a um ou outro dos semi-espacos definidos por um hiperplano (figura 4.7), por sua vez definido pelos pesos do *perceptron*.

Esta é uma severa limitação, pois em geral apenas funções muito simples são linearmente separáveis. Não é o caso da já citada função booleana ou-exclusivo

(XOR), que aparece à direita da figura 4.7.

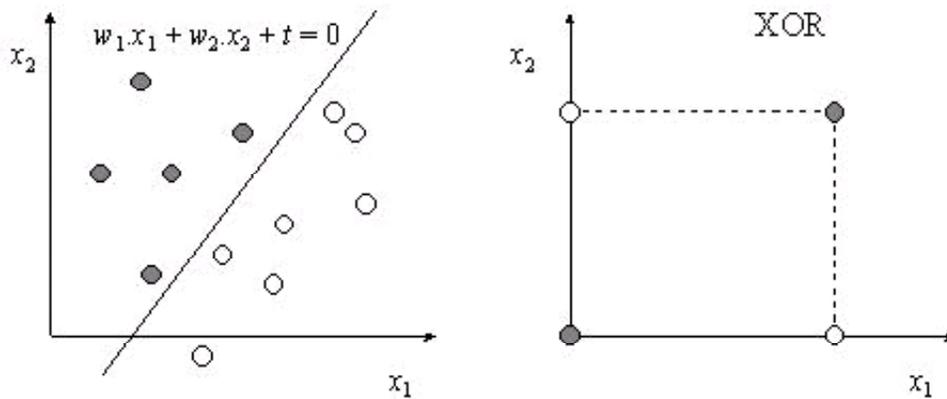


Figura 4.7: Um *perceptron* só pode distinguir entre classes linearmente separáveis, como as da esquerda. Em muitas funções isto não ocorre, como no caso da ou-exclusivo, à direita.

4.3.2 Rede Neural *Backpropagation* *Backpropagation* Padrão

Foi no início da década de 80 que se tornou popular a solução para o aprendizado em *Multi-Layer Perceptron (MLP)*. Este método de aprendizado é chamado *backpropagation* (denominando essas redes de *redes backpropagation*), pois ele se baseia na propagação dos erros a partir da camada de saída (onde eles são facilmente conhecidos devido à saída desejada apresentada) para as camadas anteriores. Pode-se entender intuitivamente o processo considerando que o erro de uma unidade oculta (ou seja, que não está na camada de saída da rede) é calculado somando-se o quanto ela contribuiu para os erros da camada seguinte. Os erros das unidades da camada seguinte são propagados para esta unidade oculta na mesma medida em que estão ligadas, ou seja, proporcionalmente aos pesos das conexões entre elas. Se a camada seguinte é a camada de saída, seu erro é conhecido; caso contrário, será uma camada oculta cujo erro poderá ser calculado da mesma maneira.

Tecnicamente, este processo é realizado simplesmente alterando o peso de cada conexão na direção que torna o erro menor para o conjunto de exemplos for-

necido. Isso é feito calculando-se a derivada do erro para cada peso e alterando-o na direção oposta à derivada. Com isto, há um movimento no espaço dos pesos para a rede em questão sempre em direção a um erro menor, até que se atinja um mínimo local.

Segue-se o algoritmo que resume o processo, em uma versão um pouco mais geral do que a apresentada intuitivamente, que fixa um determinado número de camadas. Ao invés de ter-se unidades que recebem entradas apenas das unidades das camadas anteriores, teremos uma rede em que as unidades estão em seqüência e cada uma recebe como entradas as ativações de todas as unidades anteriores. Isso inclui as unidades de saída, que recebem como entrada as ativações de todas as unidades, inclusive as das outras unidades de saída anteriores a si. Naturalmente, isso não impede que o esquema por camadas, mais comum, seja usado, pois ele é a versão particular em que as conexões que não vão de uma camada para outra estão fixadas em zero.

O Algoritmo

De acordo com Fausett (1994), o algoritmo *backpropagation* foi criado independentemente por David Parker e LenCun [Parker, 1985] [LenCun, 1986]. O treinamento de uma rede usando esse algoritmo envolve três estágios: o *feedforward* (a passagem para frente) do padrão de treinamento da entrada, o *backpropagation* (a retropropagação) do erro associado e o *ajustamento dos pesos*.

Durante o *feedforward*, cada unidade de entrada X_i recebe um sinal de entrada e envia o sinal para cada uma das unidades ocultas Z_1, \dots, Z_p . Cada unidade oculta calcula sua ativação e envia seu sinal z_j para cada uma das unidades de saída. Cada unidade de saída calcula sua ativação y_k para formar a resposta da rede para o dado padrão de entrada.

Durante o treinamento, cada unidade de saída compara sua ativação y_k calculada com o valor esperado t_k para determinar o erro associado com o padrão dessa unidade. Baseado nesse erro, o fator δ_k ($k = 1, \dots, m$) é calculado. δ_k é usado para distribuir o erro na unidade de saída Y_k de volta a todas as unidades da camada anterior (as unidades ocultas que estão conectadas a Y_k). Ele é também usado (depois) para atualizar os pesos entre a camada oculta e a camada de saída. De uma maneira similar, o fator δ_j ($j = 1, \dots, p$) é calculado para cada unidade oculta Z_j . Ele não é necessário para propagar o erro de volta para a camada de entrada, mas é usado para atualizar os pesos entre a camada oculta e a camada de entrada.

Depois dos fatores δ terem sido determinados, os pesos para todas as camadas

são ajustados simultaneamente. O ajustamento para o peso w_{jk} (da unidade oculta Z_j para a unidade de saída Y_k) é baseado no fator δ_k e na ativação z_j unidade oculta Z_j . O ajustamento para o peso v_{ij} (da unidade de entrada X_i para a unidade oculta Z_j) é baseado no fator δ_j e na ativação x_i unidade de entrada.

Nomenclatura

A nomenclatura usada no algoritmo de treinamento para a *rede backpropagation* é a seguinte:

x : Vetor de treinamento de entrada.

$$x = (x_1, \dots, x_i, \dots, x_n)$$

t : Vetor de saída esperado.

$$t = (t_1, \dots, t_k, \dots, t_m).$$

δ_k : Fator de ajustamento de peso e correção de erro para w_{jk} que é próprio para um erro na unidade de saída Y_k ; também, a informação sobre o erro na unidade Y_k que é propagada de volta à camada oculta que ativa a unidade Y_k .

δ_j : Fator de ajustamento de peso e correção de erro para v_{ij} que de saída para é próprio para a retropropagação da informação do erro da camada a unidade oculta Z_j .

α : Taxa de aprendizagem.

X_i : Unidade de entrada i :

Para cada unidade de entrada, o sinal de entrada e o sinal de saída são os mesmos, nomeado x_i .

v_{oj} : *Bias* sobre a unidade oculta j .

Z_j : Unidade oculta j .

A entrada da rede para Z_j é denotada z_in_j :

$$z_in_j = v_{oj} + \sum_i x_i v_{ij}$$

O sinal de saída (ativação) de Z_j é denotado z_j :

$$z_j = f(z_in_j)$$

w_{ok} : *Bias* sobre a unidade de saída k .

Y_k : Unidade de saída k :

A entrada da rede para Y_k é denotada y_in_k :

$$y_in_k = w_{ok} + \sum_j z_j w_{jk}$$

O sinal de saída (ativação) de Y_k é denotado y_k :

$$y_k = f(\text{in}_k)$$

Função de ativação

Uma função para uma *rede backpropagation* deveria ter várias características importantes: ela deveria ser contínua, diferenciável e não decrescente monotonicamente. Além disso, para eficiência computacional, é desejável que sua derivada seja fácil de computar. Para a função de ativação mais usada comumente, o valor da derivada (num valor particular da variável independente) pode ser expresso em termos do valor da função (no valor da variável independente). Normalmente, é esperado que a função sature, isto é, aproximar de valores máximos finitos e valores mínimos assintoticamente.

Um das funções de ativação mais comum é a função logística binária, a qual tem um intervalo de (0, 1) e é definida como:

$$f_1(x) = \frac{1}{1 + \exp(-x)},$$

com

$$f_1'(x) = f_1(x)[1 - f_1(x)].$$

Pseudo_código do Algoritmo de treinamento *backpropagation*

A função de ativação definida na seção anterior pode ser usada no algoritmo *backpropagation* descrito aqui. A forma dos dados (especialmente os valores dos objetivos) é um fator importante na escolha da função apropriada.

O algoritmo é como se segue:

Passo 0: Inicia os pesos.

(Conjunto de pequenos valores aleatórios).

Passo 1: Enquanto a condição de parada for falsa, faça passos 2 - 9.

Passo 2: Para cada par de treinamento, faça passos 3 - 8.

Feedforward (passe para frente):

Passo 3: Cada unidade de entrada ($X_i, i = 1, \dots, n$) recebe sinal de entrada x_i e envia esse sinal para todas as unidades da camada acima (camadas ocultas).

Passo 4: Cada unidade oculta ($Z_j, j = 1, \dots, p$) soma seus sinais de peso da entrada,

$$z_{in_j} = v_{oj} + \sum_{i=1}^n x_i v_{ij}$$

e aplica sua função de ativação para computar seu sinal de saída,

$$z_j = f(z_{in_j}),$$

e envia seu sinal para todas as unidades no nível acima (unidades de saída).

Passo 5: Cada unidade de saída ($Y_k, k = 1, \dots, m$) soma seus sinais de peso vindos da entrada,

$$y_{in_k} = w_{ok} + \sum_{j=1}^p z_j w_{jk}$$

e aplica sua função de ativação para computar seu sinal de saída,

$$y_k = f(y_{in_k}).$$

Retropropagação (*backpropagation*) do erro:

Passo 6: Cada unidade de saída ($Y_k, k = 1, \dots, m$) recebe um padrão de objetivo correspondente ao padrão de treinamento da entrada, calcula seu termo de informação do erro,

$$\delta_k = (t_k - y_k) f'(y_{in_k}),$$

calcula seu termo de correção de peso (usado para atualizar w_{jk} depois),

$$\Delta w_{jk} = \alpha \delta_k z_j,$$

calcula seu termo de correção de *bias* (usado para atualizar w_{ok} posteriormente),

$$\Delta w_{ok} = \alpha \delta_k,$$

e envia δ_k para as unidades no nível abaixo.

Passo 7: Cada unidade oculta ($Z_j, j = 1, \dots, p$) soma seu delta de entrada (de unidades no nível acima),

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk},$$

multiplica pela derivada de sua função de ativação para calcular seu termo de informação do erro,

$$\delta_j = \delta_{in_j} f'(z_{in_j}),$$

calcula seu termo de correção de peso (usado para atualizar v_{ij} posteriormente),

$$\Delta v_{ij} = \alpha \delta_j x_i,$$

e calcula seu termo de correção de *bias* (usado para atualizar v_{oj} posteriormente),

$$\Delta v_{oj} = \alpha \delta_j.$$

Atualizando pesos e *bias*:

Passo 8: Cada unidade de saída ($Y_k, k = 1, \dots, m$) atualiza seu *bias* e pesos ($j = 0, \dots, p$):

$$w_{jk}(\textit{novo}) = w_{jk}(\textit{velho}) + \Delta w_{jk}.$$

Cada unidade oculta ($Z_j, j = 1, \dots, p$) atualiza seu *bias* e pesos ($i = 0, \dots, n$):

$$v_{ij}(\textit{novo}) = v_{ij}(\textit{velho}) + \Delta v_{ij}.$$

Passo 9: Teste de condição de parada.

Observações importantes

O fato de cada unidade ter como entrada valores de ativação apenas de unidades anteriores é uma restrição importante. Isso permite o cálculo exato das derivadas da função do erro quadrático para um conjunto de treinamento em apenas um passo através da rede, como será mostrado adiante.

A taxa de aprendizado é geralmente um valor pequeno, para que o processo se aproxime da idealização contínua, mas se ela for pequena demais, naturalmente o processo será muito lento. Por isso, seu valor será o maior possível sem que a rede deixe de adaptar seus pesos corretamente. Esta é uma escolha geralmente empírica, que depende de cada caso.

Com este processo, são formadas representações internas úteis nas unidades ocultas da rede, representações necessárias para transformar a entrada de forma que a camada de saída possa separar as classes apresentadas linearmente. A formação destas representações depende em certo grau dos valores iniciais dos pesos das conexões, escolhidos aleatoriamente dentro de uma faixa de valores pequenos. Muitas vezes estas representações internas são mais eficientes do que inicialmente se supõe.

É interessante notar como, embora o sinal sobre o qual se calcula as derivadas seja normalmente o erro em relação a saídas desejadas, é necessário apenas que se tenha uma medida que se queira minimizar e cuja derivada possa ser calculada em relação aos pesos da rede. Uma aplicação deste princípio já foi usada para forçar a generalização da solução obtida, por exemplo, minimizando uma soma dos erros

com uma medida de simplicidade da rede.

Aqui cabem algumas observações sobre os pontos falhos do *backpropagation*. Nem sempre as representações internas necessárias são formadas, devido ao caráter aleatório dos pesos iniciais. Nestes casos, muitas vezes é necessário variar parâmetros como tamanho e taxa de aprendizado de forma *ad hoc*. Considerando o problema em um outro nível, a minimização do erro quadrático no espaço de pesos das conexões possui o inconveniente de todo método de regressão linear: pode ser atingido um mínimo local que não é o global. Além disso, não há indicações a priori sobre qual deva ser o tamanho da rede, nem quantas camadas ela deveria ter (quando usamos uma rede em camadas). Sua aplicação tem, portanto, um forte fator empírico. Ainda assim, o aprendizado por *backpropagation* é bastante eficiente e pode tratar conjuntos complexos de dados. Ele é, de longe, o mais usado em aplicações práticas, quase sempre em classificação de padrões [Braz, 1998].

Questões a serem consideradas na implementação do *backpropagation*

- **Inicialização dos pesos e *biases*:** a escolha dos pesos iniciais é muito importante, pois influencia no quão rápido a rede irá convergir, se ela atingirá um mínimo local ou global da superfície de erro. Como a atualização do peso entre dois neurônios depende da ativação de ambos, é essencial evitar a escolha de pesos iniciais que tornem nula a ativação deles. Segundo Fausett (1994), os pesos iniciais não devem ser valores muito grandes, pois o valor da derivada da função sigmóide pode ser um valor muito pequeno e o treinamento da rede se torna demorado. Por outro lado, se os pesos iniciais forem valores muito pequenos, a entrada dos neurônios escondidos ou de saída serão valores próximos de zero, o que também implica em um aprendizado muito lento. Geralmente os pesos da rede são inicializados como números aleatórios pequenos. Uma forma é inicializar os pesos como valores aleatórios no intervalo entre -1 e 1 ou -0,5 e 0,5.
- **As funções de ativação:** como mencionado anteriormente, o neurônio gera uma saída aplicando uma função na sua entrada. Geralmente, embora não seja uma regra, a mesma função de ativação é utilizada pelos neurônios da mesma camada. Redes Multi-Camadas requerem funções não-lineares, uma vez que, treinadas utilizando funções lineares, não apresentam nenhuma vantagem em relação às redes de camada única. As funções de ativação mais usadas em *redes backpropagation* são a função logística e a tangente hiperbólica. Isso se deve ao fato de o cálculo da derivada dessas funções ser

extremamente simples e não requerer nenhum esforço computacional complementar durante o processo de treinamento.

Função logística binária:

$$f(x) = \frac{1}{1 + \exp(-x)} \quad f'(x) = f(x)[1 - f(x)]$$

Função logística bipolar:

$$g(x) = \frac{2}{1 + \exp(-x)} - 1 \quad g'(x) = 0.5[1 + g(x)][1 - g(x)]$$

- **Por quanto tempo a rede deve ser treinada:** No treinamento, a rede deve adquirir a capacidade de memorização e de generalização. Desta forma, ela será capaz de produzir saídas corretas tanto para os padrões de entrada já aprendidos quanto para os nunca vistos anteriormente. Conforme Fausett (1994), para alcançar esse objetivo, a rede deve ser treinada até que o erro quadrático alcance um mínimo. Hecht-Nielsen sugere dois conjuntos de dados durante o treinamento: um de treinamento e outro para testes [HN, 1990]. Estes dois conjuntos são disjuntos. Os ajustes dos pesos são computados utilizando-se os padrões do conjunto de treinamento; entretanto, em intervalos de tempo, o erro é computado para padrões do conjunto de teste. O treinamento continua enquanto o erro para os padrões de teste diminui. No momento em que esse erro começa a aumentar, a rede está começando a memorizar os padrões de treinamento muito especificamente (perdendo a habilidade de generalizar). Neste ponto, o treinamento termina.
- **Quantos pares de treinamento são necessários:** segundo Fausett (1994), existe uma relação entre o número de pesos da rede W , o número de padrões de treinamento P e a taxa de correção da classificação e .

$$e = \frac{W}{P} \quad P = \frac{W}{e}$$

Ainda conforme Fausett (1994), uma rede treinada para classificar uma fração de $1 - \frac{e}{2}$ padrões de treinamento corretamente, onde $0 < e < \frac{1}{8}$, classificará os padrões de teste corretamente se o número de padrões de treinamento for pelo menos P (P obedece a relação descrita acima). Por exemplo, se $e = 0.1$ e $W = 70$ (rede com setenta pesos), será necessário um conjunto

de treinamento com 700 padrões ($P = \frac{70}{0.1}$) para assegurar que 90% dos padrões de teste serão classificados corretamente, assumindo que a rede foi treinada para classificar 95% dos padrões de treinamento corretamente.

- **A apresentação dos dados:** A representação dos dados influencia demasiadamente o desempenho da rede. A velocidade de aprendizado está intimamente relacionada com o formato dos dados de entrada da rede. Geralmente, os vetores de entrada e saída têm componentes com valores no mesmo intervalo. O cálculo do ajuste de um peso depende da ativação da unidade anterior. Desta forma, se ativação desta unidade é zero, a rede não irá aprender. Uma sugestão é que os dados estejam na forma bipolar e que a função de ativação seja a função logística bipolar.
- **Número de camadas escondidas:** segundo Fausett (1994), uma camada escondida é suficiente para uma *rede backpropagation* aproximar qualquer mapeamento contínuo de padrões de entrada em padrões de saída, com um certo grau de correção. Entretanto, duas camadas escondidas tornam o treinamento mais fácil e rápido em algumas situações.
- **Condições de parada:** geralmente, o algoritmo pára quando o erro quadrático calculado na saída para todos os padrões de treinamento é inferior ao erro máximo tolerado; ou quando o número de iterações (número de vezes que o algoritmo processa todos os padrões do conjunto de treinamento) excede o número de iterações máximo definido. O tempo também pode ser utilizado para definir a condição de parada da rede. Uma cota de tempo máximo é estabelecida. Quando a rede ultrapassa essa cota, o algoritmo pára.

4.3.3 Rede Neural *Backpropagation* Variações

Várias modificações podem ser feitas para o algoritmo *backpropagation* apresentado na seção anterior, as quais devem melhorar seu desempenho em algumas situações. Essas modificações envolvem principalmente o procedimento de atualização dos pesos. A seguir, serão apresentados alguns métodos alternativos para atualização dos mesmos.

Momentum

No algoritmo *backpropagation com momentum*, a atualização dos pesos é feita na direção de uma combinação entre o gradiente atual e o gradiente anterior. De acordo com Fausett (1994), esta técnica apresenta vantagens quando algum padrão de treinamento for muito diferente da maioria dos outros padrões. É desejável usar uma taxa de aprendizagem pequena para não causar uma mudança brusca da direção do aprendizado quando alguns padrões de treinamento forem muito diferentes dos demais. Entretanto, se os padrões forem relativamente similares também é interessante manter a aprendizagem num ritmo mais acelerado. Em algumas aplicações, a convergência é mais rápida quando é adicionado um termo *momentum* na fórmula de atualização dos pesos. Para isso, é necessário salvar os pesos de um ou mais padrões de treinamento processados anteriormente. Por exemplo, na forma mais simples do algoritmo *backpropagation com momentum*, o novo peso depende dos pesos dos passos t e $t - 1$. As fórmulas para atualizar os pesos para o algoritmo *backpropagation com momentum* são:

$$w_{jk}(t + 1) = w_{jk}(t) + \alpha \delta_k z_j + \mu [w_{jk}(t) - w_{jk}(t - 1)]$$

ou

$$\Delta w_{jk}(t + 1) = \alpha \delta_k z_j + \mu \Delta w_{jk}(t)$$

e

$$v_{ij}(t + 1) = v_{ij}(t) + \alpha \delta_j x_i + \mu [v_{ij}(t) - v_{ij}(t - 1)]$$

ou

$$\Delta v_{ij}(t + 1) = \alpha \delta_j x_i + \mu \Delta v_{ij}(t)$$

onde μ é o parâmetro do *momentum* ($0 \leq \mu \leq 1$).

O *momentum* permite à rede fazer grandes ajustes nos pesos quando as correções para muitos padrões de treinamento estão na mesma direção. Entretanto, segundo Jacobs (1988), a técnica do *momentum* apresenta algumas limitações, entre elas, o fato de o peso poder ser atualizado numa direção que aumenta o erro.

Atualização dos pesos em lote

Às vezes, é vantajoso acumular os termos de correção dos pesos de muitos padrões e posteriormente fazer uma única atualização (seria igual à média aritmética dos termos de correção dos pesos acumulados) para cada peso, em vez de atualizar os pesos sempre que um padrão é apresentado à rede.

Taxas de aprendizagem adaptativas

O algoritmo *backpropagation* padrão modifica os pesos na direção que a superfície de erro diminui mais rapidamente para os pesos atuais. Em geral, essa técnica não move os pesos diretamente para a direção do vetor de pesos ótimo [Fausett, 1994].

Um método desenvolvido para melhorar a velocidade de aprendizagem consiste em mudar o valor da taxa de aprendizagem durante o treinamento. Este método é aplicado em alguns problemas específicos, como por exemplo, um problema de classificação onde o conjunto de treinamento apresenta bem menos padrões de uma determinada classe do que de outras. Se não é possível a duplicação ou criação de cópias com ruídos, a taxa de aprendizagem deve aumentar quando padrões de treinamento da classe com menor número de padrões for apresentados à rede [Fausett, 1994].

Um outro método baseado na mudança do valor da taxa de aprendizagem é denominado *delta-bar-delta*. Este algoritmo permite que a taxa de aprendizagem de cada peso mude ao longo do processo de treinamento. Se a atualização do peso foi feita na mesma direção por muitas iterações (o sinal da derivada parcial do erro com relação ao peso permaneceu o mesmo por muitas iterações), então o valor de sua taxa de aprendizagem aumenta. Entretanto, se a direção do ajuste do peso alternou por muitas iterações (o sinal da derivada parcial do erro alternou por muitas iterações), o valor de sua taxa de aprendizagem diminui. Não é uma garantia que estas heurísticas sempre melhorem o desempenho da rede, embora elas tenham feito isso em muitos exemplos.

A regra *delta-bar-delta* modifica os pesos da seguinte forma:

$$w_{jk}(t+1) = w_{jk}(t) + \alpha_{jk}(t+1) \left(\frac{\partial E}{\partial w_{jk}} \right) = w_{jk} + \alpha_{jk}(t+1) \delta_k z_j$$

Este é o ajuste do peso para o algoritmo *backpropagation* padrão, mas agora o peso pode ser modificado por uma proporção diferente da derivada parcial do erro com relação a esse peso.

Para as unidades de saída, define-se um delta:

$$\Delta_{jk} = \left(\frac{\partial E}{\partial w_{jk}} \right) = -\delta_k z_j$$

Para as unidades escondidas:

$$\Delta_{ij} = \left(\frac{\partial E}{\partial v_{ij}} \right) = -\delta_j x_i$$

A regra *delta-bar-delta* utiliza uma combinação da derivada parcial atual e a anterior para formar "delta-barr" para cada unidade de saída:

$$\bar{\Delta}_{jk}(t) = (1 - \beta)\Delta_{jk}(t) + \beta\bar{\Delta}_{jk}(t - 1)$$

e para cada unidade escondida:

$$\bar{\Delta}_{ij}(t) = (1 - \beta)\Delta_{ij}(t) + \beta\bar{\Delta}_{ij}(t - 1)$$

O valor do parâmetro β deve ser especificado pelo usuário ($0 < \beta < 1$).

A heurística de que a taxa de aprendizagem deve aumentar se os ajustes dos pesos foram feitos na mesma direção por várias iterações sucessivas é implementada elevando-se a taxa de aprendizagem se $\bar{\Delta}_{jk}(t - 1)$ e $\bar{\Delta}_{jk}(t)$ forem do mesmo sinal. Por outro lado, a taxa de aprendizagem deve ser reduzida se $\bar{\Delta}_{jk}(t - 1)$ e $\bar{\Delta}_{jk}(t)$ possuírem sinais opostos. A nova taxa de aprendizagem é obtida por:

$$\alpha_{jk}(t + 1) \begin{cases} \alpha_{jk}(t) + \kappa & \text{se } \bar{\Delta}_{jk}(t - 1)\Delta_{jk}(t) > 0, \\ (1 - \gamma)\alpha_{jk}(t) & \text{se } \bar{\Delta}_{jk}(t - 1)\Delta_{jk}(t) < 0, \\ \alpha_{jk}(t) & \text{caso contrário.} \end{cases}$$

Os valores dos parâmetros κ e γ devem ser especificados pelos usuários.

Jacobs (1988) fez alguns testes com o algoritmo *backpropagation* padrão e as variações apresentadas nessa seção. Para resolver a função lógica *Xor*, foram realizadas vinte e cinco simulações, utilizando-se uma rede com duas unidades de entrada, duas unidades escondidas e uma unidade de saída (a rede foi inicializada com pesos distintos). A rede *Xor* recebia entradas binárias e os valores desejados na saída eram 0, 1 e 0, 9. As simulações usaram a técnica da atualização dos pesos em lote. As condições de parada consideradas foram erro total ao quadrado $\leq 0,04$ e uma média de mais de 50 épocas. Os valores usados para os parâmetros estão ilustrados na tabela 4.1:

Tabela 4.1: Valores para os parâmetros nos testes de Jacobs (1988).

	A	μ	κ	γ	β
<i>Backpropagation</i>	0.1				
<i>Backpropagation com momentum</i>	0.75	0.9			
<i>Delta-bar-delta</i>	0.8		0.035	0.333	0.7

Os valores dos testes realizados podem ser observados na tabela 4.2. Eles mostraram que, embora o treinamento utilizando *backpropagation* com a regra *delta-bar-delta* nem sempre converge (22 conversões em 25 simulações), quando ele conseguir convergir, ele o faz de forma muito rápida em comparação com os demais métodos.

Tabela 4.2: Valores dos testes realizados por Jacobs (1988).

Método	Simulações	Sucessos	Média de Épocas
<i>Backpropagation</i>	25	24	16,859.8
<i>Backpropagation com momentum</i>	25	25	2,056.3
<i>Delta-bar-delta</i>	25	22	447.3

4.3.4 Outros modelos para redes neurais

Há muitas variações possíveis para redes conexionistas, na verdade tantas que não é possível seu agrupamento em uma única formalização teórica. Isto torna difícil o estudo de suas propriedades de modo formal.

De qualquer forma, pode-se identificar variações mais comuns e importantes na literatura. Algumas delas são presença ou não de unidades ocultas, presença ou não de laços na rede (recorrência), utilização de aprendizado supervisionado ou não supervisionado, unidades bipolares ou binárias.

Embora todas estas variações sejam importantes, apenas a presença de unidades escondidas realmente altera significativamente o conjunto de problemas que podemos abordar com as redes conexionista. Sem elas, é impossível tratar problemas não linearmente separáveis.

4.4 Aplicações

Existem aplicações de redes neurais para solucionar diversos problemas, cuja resolução é inviável utilizando-se computação convencional.

As principais aplicações em redes neurais estão relacionadas com o reconhecimento de padrões, a classificação e mapeamento de dados. Como exemplo de aplicações, pode-se citar: reconhecimento de padrões em linhas de montagem, classificação de dados, controle de processos e robótica, análise e processamento de sinais, análise de imagens, análise de voz, diagnóstico médico, análise e di-

agnóstico de mercados financeiros, análise de assinatura radar e sonar, análise de odores e compostos gasosos.

Capítulo 5

Metodologia e Resultados Obtidos

5.1 Introdução

Redes neurais artificiais têm sido utilizadas para analisar padrões de odor e vêm mostrando resultados promissores. Grande parte das abordagens já propostas para o sistema de reconhecimento de padrões dos narizes artificiais fazem uso da conhecida rede *Multi-Layer Perceptron (MLP)* treinada com o algoritmo *Backpropagation* [Yamazaki, 2001]. Existem diversos outros tipos de redes neurais que podem ser testados para o mesmo fim, produzindo resultados que podem ser comparados com os obtidos por uma rede MLP aplicada aos mesmos padrões de entrada.

Oliveira (2002) apresenta um sistema de reconhecimento de padrões de odores de café, utilizando redes MLP treinadas com *backpropagation*. Seu sistema apresentou resultados promissores, mas os padrões de entrada foram insuficientes para obter resultados mais significativos, devido à forma como as redes foram implementadas.

Neste capítulo, é proposta uma reformulação do sistema de Oliveira (2002), utilizando o algoritmo de aprendizagem *backpropagation* padrão e *backpropagation com momentum* para o treinamento das redes neurais e uma redução do número de neurônios de entrada, através de um método de pré-processamento dos dados diferente do que foi usado em seus experimentos.

Foram analisados os sinais gerados por um conjunto de doze sensores químicos, expostos aos odores de três tipos distintos de café. Os detalhes do sistema de reconhecimento e do pré-processamento são apresentados a seguir, bem como os

resultados obtidos para diferentes números de unidades ocultas em redes com uma camada intermediária.

5.2 Descrição do Problema e da Base de Dados

O problema em questão trata da classificação de amostras de café em três tipos de bebida: dura, mole ou rio. A base de dados utilizada foi obtida pelo departamento de química da Universidade Federal de Lavras, através de um nariz artificial composto de doze sensores químicos. Amostras de café foram analisadas com a microbalança de quartzo e a aquisição dos dados foi feita de forma automática utilizando-se um computador.

Foram realizadas três repetições de aquisição de dados. Em cada uma destas repetições, para cada um dos três tipos de bebida, o valor da resistência de cada um dos doze sensores foi registrado a cada 0,05 minutos, sendo que o processo durava vinte um minutos. Dessa forma, cada sensor obteve 420 valores registrados para cada tipo de café.

O conjunto formado pelos doze valores dos doze sensores, no mesmo instante de tempo, foi considerado como um padrão da base de dados. Assim, cada repetição contém 1260 padrões (420 de cada tipo de café). Como existem três repetições, tem-se um total de 3780 padrões na base de dados.

A tabela 5.1 traz um exemplo de uma das repetições de aquisição de dados para o tipo de café bebida dura, permitindo uma melhor visualização.

Tabela 5.1: Exemplo de uma das repetições de aquisição de dados do café bebida dura.

Tempo (min)	S1	S2	S2	S4	...	S12
0	9985668	9988302	9943967	9991708	...	9991200
0.05	9985668	9988298	9943967	9991707	...	9991200
0.1	9985668	9988297	9943966	9991702	...	9991200
0.15	9985668	9988298	9943966	9991707	...	9991200
...
20.9	9985621	9988238	9943932	9991668	...	9991189
20.95	9985621	9988237	9943932	9991667	...	9991188

5.3 Experimentos com *Multi-Layer Perceptron* (MLP)

5.3.1 Divisão do Conjunto de Padrões em Treinamento e Teste

O conjunto de 3780 padrões foi dividido em dois conjuntos: o de treinamento da rede neural, contendo $\frac{2}{3}$ da quantidade total de padrões e o de teste, contendo $\frac{1}{3}$ restante. Vale ressaltar que o conjunto de treinamento possui 50% dos padrões de cada tipo de café nas duas primeiras aquisições (para cada tipo, 420 padrões são escolhidos aleatoriamente entre os 840 existentes nas aquisições 1 e 2). Para o conjunto de teste, são usados os 420 padrões restantes. Assim, os três tipos de café estão representados pela mesma quantidade de padrões em cada um dos três conjuntos de dados.

5.3.2 Pré-processamento dos Dados

O pré-processamento dos dados é de vital importância para o êxito da rede na resolução do problema abordado, pois o desempenho do treinamento depende do formato dos dados de entrada. Yamasaki (2001) apresenta várias técnicas que podem ser usadas para esse pré-processamento. A idéia é encontrar dados relativos aos dados originados dos sensores, pois estes são apenas valores sem muito significado para rede neural.

A proposta de Oliveira (2002) para o pré-processamento, ao implementar um sistema de reconhecimento de odores de café, foi encontrar a derivada do polinômio interpolador dos dados de entrada (que são os pontos (tempo, Freq), onde Freq é a frequência do sensor s), a velocidade máxima (v_{max}) da curva de adsorção e o tempo (t_{90}) em que a frequência é igual a 90% da frequência máxima. Dessa forma, os dados de entrada da rede foram constituídos pelos coeficientes da derivada, v_{max} e t_{90} calculados para os dados de cada sensor. Como o polinômio interpolador calculado era de grau seis, por apresentar um ajuste muito bom dos pontos experimentais, sua derivada apresenta seis coeficientes e somando-se v_{max} e t_{90} tem-se o total de oito entradas por sensor. Assim a entrada da rede (padrão de entrada) era constituída por 12 (n^0 de sensores) $\times 8 + 1$ (temperatura) = 97 valores.

Para diminuir o número de valores da entrada da rede proposta por Oliveira (2002), foi implementado, para os experimentos deste trabalho, o mesmo pré-processamento utilizado por Yamasaki (2001), ao implementar um sistema de reconhecimento de odores de três safras distintas de um mesmo vinho.

Yamasaki (2001) afirma que, muitas vezes, é necessária uma normalização dos dados, para compensar as flutuações de concentração. A discriminação entre

odores complexos similares pode ser melhorada substancialmente se os comprimentos dos vetores de resposta forem normalizados. Em seus experimentos com MLP, é realizada uma normalização dos valores obtidos de seis sensores distintos de polímero condutor, utilizando a expressão apresentada logo abaixo.

$$x_{norm} = \frac{2x - x_{max} - x_{min}}{x_{max} - x_{min}}$$

onde x_{norm} é o valor normalizado correspondente ao valor original x de resistência, e x_{min} e x_{max} são, respectivamente, os valores mínimo e máximo entre os valores de resistência correspondentes ao sensor em questão na base de dados utilizada.

O conjunto de padrões que constitui a entrada das redes utilizadas neste trabalho, foi, então, normalizado de acordo com a expressão apresentada por Yamasaki (2001). Os valores foram normalizados separadamente para cada sensor de modo a ficar compreendido na faixa entre -1 e $+1$, pois as unidades das redes neurais utilizaram função de ativação logística bipolar.

5.3.3 O Sistema de Reconhecimento de Padrões

As redes implementadas por Oliveira (2002) apresentavam 97 unidades de entrada (definidas pelo pré-processamento empregado), 60 unidades para uma camada intermediária e 70 para duas camadas intermediárias (definidas empiricamente) e 3 unidades de saída (cada uma correspondendo a um tipo de bebida). Em seus experimentos, foi concluído que a quantidade de padrões de treinamento disponível era insuficiente para treinar a rede de forma adequada, ou seja, para se obter um resultado indicando que porcentagem do conjunto de teste a rede foi capaz de classificar corretamente após seu treinamento.

Para reformular o sistema de Oliveira (2002), foram implementadas redes neurais MLP, com as seguintes configurações:

- uma camada intermediária, onde o número de unidades é definido empiricamente;
- a camada de entrada possui 12 unidades, onde cada unidade recebe um dos valores pré-processados correspondentes ao padrão de entrada;
- a camada de saída tem 3 unidades, cada uma correspondendo a um tipo de café, pois foi utilizada a codificação de saída *1-of-m*, onde m é o número de classes.

As unidades da camada intermediária e de saída possuem função de ativação logística bipolar. A rede contém todas as possíveis conexões *feedforward* entre camadas adjacentes, sem possuir conexões que ligam unidades de camadas distintas não-adjacentes. Os algoritmos de treinamento usados foram o *backpropagation* padrão, já implementado por Oliveira (2002) e o *backpropagation com momentum*.

Foram treinadas três topologias diferentes (4, 8, e 20 unidades na camada intermediária) para que os resultados obtidos fossem comparados entre si.

Detalhes de implementação

- Para implementação das rede neurais e do pré-processamento foi utilizada a linguagem de programação c++.
- O(s) arquivo(s) texto resultantes do pré-processamento constituem a entrada da rede.
- Cada camada, com exceção da camada de saída, possui um *bias*.
- Os parâmetros que determinam a condição de parada do programa são o erro quadrático máximo tolerado e o número máximo de iterações. Quando o erro quadrático calculado para todos os padrões de treinamento for menor ou igual ao erro quadrático máximo e o número de iterações executadas for maior que o número máximo de iterações, o programa pára e o treinamento é finalizado.
- Inicialmente são passados para o programa alguns dados para que a rede seja montada. Estes dados podem estar salvos em um arquivo texto ou podem ser passados para o programa via teclado. Ao ser executado o programa, o usuário deve optar por uma dessas alternativas. Se for escolhida a primeira, ele deve informar ao programa qual o nome do arquivo texto com as configurações da rede. Senão, serão pedidos pelo programa: o número de camadas da rede (incluindo a camada de entrada e a camada de saída), o número de neurônios de cada uma dessas camadas, a taxa de aprendizagem (valor entre 0 e 1), o erro quadrático máximo tolerado e o número máximo de iterações (parâmetros para determinar quando o programa pára). Na rede treinada com *backpropagation com momentum*, ainda é pedido o valor do fator *momentum* (valor entre 0 e 1).
- Após o treinamento, o usuário pode optar por salvar as configurações atuais da rede em um arquivo texto e por executar a rede já treinada.

5.3.4 Resultados Obtidos com MLP usando *Backpropagation* Padrão

A rede utilizada foi treinada com uma taxa de aprendizagem de 0,02, erro quadrático máximo tolerado de 0,05 e número de iterações máxima igual a 100.

Para cada topologia considerada, apresenta-se uma tabela contendo os aspectos observados nas execuções realizadas.

A *iteração principal* mostrada nas tabelas envolve 420 iterações menores, isto é, uma para cada *padrão* de entrada. O *erro quadrático calculado no treinamento* é válido para cada padrão da base de dados, ou seja, a cada doze valores oriundos dos sensores, é calculado um erro de acordo com a saída desejada em cada unidade de saída.

4 unidades intermediárias

Tabela 5.2: Resultados obtidos para MLP com 4 unidades escondidas usando *backpropagation* padrão.

	Iteração Principal	Padrão	Erro quadrático no treinamento
bebida dura	1	1	0.563593
		2	0.545761
		3	0.526138
		10	0.429519
		100	0.137471
		420	0.0363316
	2	1	0.0482741
		2	0.0460972
		3	0.0449052
		10	0.0320521
		100	0.0232838
		420	0.0103725
bebida mole	1	1	0.694171
		2	0.678341
		3	0.656896
		10	0.543009
		100	0.178856
		420	0.0363316
	2	1	0.0254485
		2	0.0249247
		3	0.0235555
		10	0.0200763
		100	0.0195116
		420	0.012041
bebida rio	1	1	0.587551
		2	0.577511
		3	0.568431
		10	0.488653
		100	0.259527
		420	0.0283462
	2	1	0.0251149
		2	0.0250972
		3	0.0250013
		10	0.0238991
		100	0.0208032
		420	0.0110842

8 unidades intermediárias

Tabela 5.3: Resultados obtidos para MLP com 8 unidades escondidas usando *backpropagation* padrão.

	Iteração Principal	Padrão	Erro quadrático no treinamento
bebida dura	1	1	0.320356
		2	0.302323
		3	0.293117
		10	0.239732
		100	0.135107
		420	0.0298263
	2	1	0.0390426
		2	0.0354128
		3	0.0351013
		10	0.0345093
		100	0.0220691
		420	0.0143743
bebida mole	1	1	0.521897
		2	0.500004
		3	0.488228
		10	0.419718
		100	0.128859
		420	0.0431323
	2	1	0.0253541
		2	0.0225439
		3	0.0251593
		10	0.0229479
		100	0.0122352
		420	0.0104894
bebida rio	1	1	0.307603
		2	0.302325
		3	0.293741
		10	0.232325
		100	0.178896
		420	0.0184048
	2	1	0.024789
		2	0.0251705
		3	0.0261271
		10	0.0234231
		100	0.00972758
		420	0.00755224

20 unidades intermediárias

Tabela 5.4: Resultados obtidos para MLP com 20 unidades escondidas usando *backpropagation* padrão.

	Iteração Principal	Padrão	Erro quadrático no treinamento
bebida dura	1	1	0.277577
		2	0.262976
		3	0.248067
		10	0.198512
		100	0.119747
		420	0.0159848
	2	1	0.0281688
		2	0.02552
		3	0.0231495
		10	0.0223687
		100	0.00891315
		420	0.00747841
bebida mole	1	1	0.495377
		2	0.467286
		3	0.449511
		10	0.290659
		100	0.108215
		420	0.0234295
	2	1	0.0470093
		2	0.042569
		3	0.0499407
		10	0.038628
		100	0.00790226
		420	0.00916889
bebida rio	1	1	0.861375
		2	0.807148
		3	0.735921
		10	0.444305
		100	0.139923
		420	0.0196289
	2	1	0.0271285
		2	0.0269492
		3	0.026885
		10	0.0247052
		100	0.00922368
		420	0.00778972

Como pode ser visto nas tabelas anteriores, o erro quadrático calculado diminui significativamente a cada padrão de entrada em todas as topologias testadas, não necessitando mais que duas iterações principais do algoritmo para que a rede aprendesse. Aumentando o número de unidades na camada intermediária, o erro quadrático calculado torna-se menor para a maioria dos padrões de entrada.

Apesar do bom desempenho das redes durante o treinamento, os resultados dos testes de classificação não foram tão promissores. As redes foram capazes de calcular os pesos adequados entre as ligações para mapear as entradas de um certo tipo de bebida para a saída desejada, porém, ao executar a rede treinada, ela não era capaz de diferenciar entre os três tipos de bebida. Para qualquer padrão de entrada, a resposta era sempre o tipo de bebida para o qual ela foi treinada. Por exemplo, se a rede fosse treinada para classificar bebida dura, ao receber um padrão de entrada para bebida mole, ela respondia ser bebida dura.

Elaine Rich (1993) explica que, mesmo a retropropagação mostrando-se promissora como mecanismo de generalização, existem algumas armadilhas (figura 5.1). Durante a primeira parte do treinamento, o desempenho do conjunto de treinamento melhora quando a rede ajusta seus pesos através da retropropagação. Depois de um certo tempo, o desempenho da rede atinge um platô, à procura de um caminho que leve a uma melhoria ainda maior. Finalmente, esse percurso é encontrado e o desempenho do conjunto de treinamento melhora de novo. Mas o desempenho do conjunto de teste piora. Isso acontece porque a rede começa a memorizar os pares entrada-saída em vez de procurar pesos que descrevam de modo geral o mapeamento de todos os casos. Com milhares de pesos de valores reais à sua disposição, a retropropagação é teoricamente capaz de armazenar todo o conjunto de treinamento; com suficientes unidades ocultas, o algoritmo pôde aprender a atribuir uma unidade oculta a todo padrão de entrada diferente presente no conjunto de treinamento. É um atestado à força da retropropagação que isto realmente ocorra na prática.

Rich (1993) ainda afirma que há várias maneiras de impedir que a retropropagação recorra a um esquema de consulta a tabelas. Uma delas é interromper o treinamento quando o platô for atingindo, com base na suposição de que qualquer outra melhoria virá através de "trapaça". Uma outra maneira é acrescentar deliberadamente pequenas quantidades de ruído às entradas de treinamento. O ruído deve ser suficiente para impedir a memorização, mas ele não deve ser muito, para não confundir o classificador. Uma terceira maneira de ajudar a generalização é reduzir o número de unidades ocultas na rede, criando um estrangulamento entre as camadas de entrada e de saída. Confrontada com esse estrangulamento, a rede

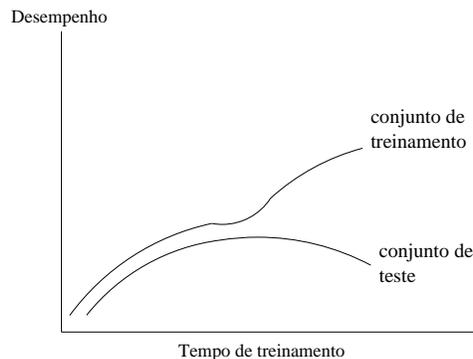


Figura 5.1: Um efeito da generalização comum em aprendizagem de redes neurais.

será forçada a apresentar representações internas compactas das entradas. Infelizmente, devido ao curto tempo para os testes da rede, não foi possível encontrar a solução correta para o problema da generalização.

5.3.5 Resultados obtidos com MLP usando *Backpropagation com Momentum*

De acordo com Fausett (1994), em algumas aplicações, a convergência é mais rápida quando é adicionado um termo *momentum* na fórmula de atualização dos pesos no algoritmo *backpropagation*.

Para este sistema, foi utilizada a mesma base de dados que gerou os resultados da seção anterior, permitindo uma comparação entre as abordagens.

Também foi usada a mesma divisão do conjunto de treinamento e teste do sistema anterior e a mesma normalização adotada para os experimentos com redes MLP usando *backpropagation* padrão, de modo que os valores ficaram compreendidos na faixa entre -1 e 1 .

A rede utilizada foi treinada usando *backpropagation com momentum*, com uma taxa de aprendizagem de $0,02$; erro quadrático máximo tolerado de $0,05$; fator *momentum* de $0,9$ e número de iterações máxima igual a 100 .

As tabelas, para cada topologia considerada, contendo os resultados observados nas execuções, são apresentadas abaixo.

4 unidades intermediárias

Tabela 5.5: Resultados obtidos para MLP com 4 unidades escondidas usando *backpropagation com momentum*.

	Iteração Principal	Padrão	Erro quadrático no treinamento
bebida dura	1	1	0.662521
		2	0.637821
		3	0.613121
		10	0.462078
		100	0.139595
		420	0.0664357
	2	1	0.0465033
		2	0.0460753
		3	0.0438049
		10	0.0358899
		100	0.0134468
		420	0.0161231
bebida mole	1	1	0.241436
		2	0.239255
		3	0.230582
		10	0.193867
		100	0.0813717
		420	0.0485209
	2	1	0.0326535
		2	0.033126
		3	0.0343889
		10	0.0279477
		100	0.0112781
		420	0.0101093
bebida rio	1	1	0.466205
		2	0.453473
		3	0.442557
		10	0.360859
		100	0.195578
		420	0.0553295
	2	1	0.0394539
		2	0.0368933
		3	0.035853
		10	0.0237469
		100	0.0115902
		420	0.00882991

8 unidades intermediárias

Tabela 5.6: Resultados obtidos para MLP com 8 unidades escondidas usando *backpropagation com momentum*.

	Iteração Principal	Padrão	Erro quadrático no treinamento
bebida dura	1	1	0.357662
		2	0.351093
		3	0.342316
		10	0.288081
		100	0.196879
		420	0.0561256
	2	1	0.028795
		2	0.0291521
		3	0.024795
		10	0.0358899
		100	0.011391
		420	0.0139638
bebida mole	1	1	0.808466
		2	0.761313
		3	0.722947
		10	0.495405
		100	0.226542
		420	0.0445932
	2	1	0.0475059
		2	0.0445119
		3	0.045657
		10	0.0361477
		100	0.03104
		420	0.0176175
bebida rio	1	1	0.48588
		2	0.449811
		3	0.42457
		10	0.287488
		100	0.163508
		420	0.0718708
	2	1	0.0453655
		2	0.0422389
		3	0.0383225
		10	0.0294759
		100	0.0162316
		420	0.0187193

20 unidades intermediárias

Tabela 5.7: Resultados obtidos para MLP com 20 unidades escondidas usando *backpropagation* com *momentum*.

	Iteração Principal	Padrão	Erro quadrático no treinamento
bebida dura	1	1	0.586319
		2	0.522976
		3	0.488067
		10	0.268512
		100	0.129747
		420	0.0259848
	2	1	0.0481688
		2	0.04552
		3	0.0431495
		10	0.0223687
		100	0.01891315
		420	0.0095984
bebida mole	1	1	0.225377
		2	0.207286
		3	0.179511
		10	0.100659
		100	0.098215
		420	0.0334295
	2	1	0.0420033
		2	0.0405245
		3	0.0349507
		10	0.028628
		100	0.00790226
		420	0.00516889
bebida rio	1	1	0.552379
		2	0.4637142
		3	0.384924
		10	0.1984305
		100	0.156762
		420	0.0168289
	2	1	0.0471545
		2	0.0469432
		3	0.042885
		10	0.0347052
		100	0.00922764
		420	0.00648972

As redes MLP treinadas com *backpropagation com momentum* não apresentaram diferenças significativas em relação às redes treinadas com *backpropagation* padrão. O erro quadrático calculado comportou-se da mesma forma apresentada nas tabelas das redes treinadas com *backpropagation* padrão e a classificação apresentou os mesmos problemas também.

Para todas as topologias, observa-se que o tempo de execução foi pequeno (ordem de segundos) e o treinamento gerou bons resultados, com uma quantidade máxima de iterações relativamente pequena. Este aspecto é muito importante em experimentos de classificação de odores, já que o objetivo é implementar um sistema eficiente de reconhecimento de padrões em um nariz artificial, que, em muitos casos, precisa optar com rapidez para detectar e identificar odores nas mais diversas aplicações. No entanto, a classificação não foi eficiente, pois a rede não apresentou generalização para os pares entrada-saída.

Capítulo 6

Conclusões e Perspectivas Futuras

6.1 Conclusões

O desenvolvimento de alternativas de classificação do café mais confiáveis que a "prova de xícara" traz mais segurança para a sua comercialização. O uso de narizes artificiais na cafeicultura pode facilitar essa classificação.

O desempenho do sistema de reconhecimento de padrões é de fundamental importância no funcionamento destes dispositivos. Nesse contexto, têm merecido atenção crescente as redes neurais artificiais, representando uma abordagem bastante adequada ao problema do processamento de sinais de odor.

Diversas abordagens com redes MLP treinadas com *backpropagation* padrão já foram propostas, porém ainda existem vários tipos de redes que podem ser testadas para a mesma finalidade, a fim de gerar resultados comparáveis com os obtidos pela rede MLP.

O sistema proposto, utilizou outra técnica para o pré-processamento e apresentou redes MLP, com topologias diferentes, treinadas com *backpropagation* padrão e com *backpropagation com momentum*. Apesar dos bons resultados obtidos durante o treinamento, as redes não obtiveram generalização e não foram capazes de classificar corretamente os padrões de entrada.

Os resultados obtidos neste trabalho podem ser úteis para futuras comparações com novos sistemas propostos para reconhecer odores.

6.2 Perspectivas Futuras

Entre as perspectivas futuras, está a possibilidade de utilizar outras abordagens de redes neurais artificiais para classificar os mesmos dados utilizados neste trabalho e outras maneiras para o pré-processamento dos sinais gerados pelos sensores. Diversos aspectos podem ser comparados, como o erro de classificação e a quantidade necessária de iterações para obter resultados considerados satisfatórios.

Outra possibilidade é o uso de outras variações do *backpropagation* para treinar a rede.

Segundo Yamasaki (2001), uma outra opção é a otimização das redes neurais utilizadas, através de técnicas como, por exemplo, algoritmos genéticos. Outra idéia, é estudar a viabilidade da implementação em hardware do sistema proposto para o reconhecimento de odores, tendo em vista a miniaturização do nariz artificial.

Referências Bibliográficas

- [Amorim, 1978] AMORIM, H. V. "Aspectos bioquímicos e histoquímicos do grão de café verde relacionados com a deterioração de qualidade". Piracicaba: ESALQ, 1978 (Tese - livre docência).
- [Barreto, 2001] BARRETO, Jorge Muniz. "Inteligência Artificial - No Limiar do Século XXI". 3. Edição, Florianópolis, 2001.
- [Braz, 1998] BRAZ, Rodrigo de Salvo. "Alto Nível em Redes Neurais". 1998. 87 págs. Dissertação - Instituto Militar de Engenharia.
- [Buck et al., 1965] BUCK, T. M., ALLEN, F. G. e DALTON, M. (1965) "*Detection of chemical species by surface effects on metals and semiconductors*". In T. Bregman e A. Dravnieks (Eds.), *Surface Effects in Detection*, Spartan Books Inc., USA, 1965.
- [Carvalho et al., 1994] CARVALHO, V.D. de.; CHAGAS, S. J. de R.; CHALFOUN, S. M.; BOTREL, N.; JUSTE JUNIOR, E. S. G. "Relação entre a composição físico-química e química do grão beneficiado e qualidade de bebida do café". Pesquisa Agropecuária Brasileira, Brasília, v. 29, n.3, p. 449-454, mar. 1994.
- [Dravnieks e Trotter, 1965] Dravnieks, A. e Trotter, P. J. (1965). "*Polar vapour detection based on thermal modulation of contact potentials*". In *J. Sci. Instrum.*, 42: 624.
- [Fausett, 1994] FAUSETT, Laurene V. "*Fundamentals of neural networks: architectures, algorithms, and applications*". 1994.
- [Gardner e Hines, 1997] Gardner, J. W. e Hines, E. L. (1997). "*Pattern Analysis Techniques*". In E. Kress-Rogers (Ed.), *Handbook of Biosensors and Electronic Noses: Medicine, Food and the Environment* (pp 633-652). CRC Press.

- [Graner, 1967] GRANER, E. A. & GODOY JÚNIOR, C.. "Manual do Cafeicultor". Ed. Universidade de São Paulo. 1967. 320 p.
- [Hebb, 1949] HEBB, D. O. *"The Organization of Behavior"*. New York: John Wiley & Sons. Introduction and Chapter 4 reprinted in Anderson & Rosenfeld [1988], pp. 45-56.
- [HN, 1990] HECHT - NIELSEN, R. *"Neurocomputing"*. Reading, MA: Addison - Wesley. 1990.
- [Ikegami e Kaneyasu, 1985] IKEGAMI, A. e KANEYASU, M. (1985). *"Olfactory detection using integrated sensors"*. Proc. 3rd Int. Conf. Solid-State Sensors and Actuators (Transducers'85), Philadelphia, PA, USA, pp. 136-139.
- [Jacobs, 1988] JACOBS, R. A. *"Increased Rates of Convergence Through Learning Rate Adaptation"*. *Neural Networks*, 1(4): 295 - 307.
- [Junior, 1995] JUNIOR, Luiz Gonzaga de Castro. "Mudança tecnológica na cafeicultura mineira: um estudo no período de 1978 a 1993". 1995. 56 p. Dissertação (Mestrado em Administração Rural) - Universidade Federal de Lavras, Lavras / MG.
- [Keller et al., 1995] KELLER, P. E., KANGAS, L. J., LIDEN, L. H., HASHEN, S. e KOUZES, R. T. *"Electronic noses and their applications"*. In *IEEE Northcon Technical Applications Conference (TAC'95)*, Portland, OR. 1995.
- [Kovacs, 1997] KOVÁCS, Zsolt László. "O Cérebro e a sua mente: uma introdução à neurociência computacional". São Paulo: Edição Acadêmica, 1997.
- [LenCun, 1986] LE CUN, Y. *"Learning Processes in an Asymmetric Threshold Network"*. In E. Bienenstock, F. Fogelman-Souli, & G. Weisbuch, eds. *Disordered Systems and Biological Organization*. NATO ASI Series, F20, Berlin: Springer-Verlag.
- [McCulloch & Pitts, 1943] McCULLOCH, W. S., & W. PITTS. (1943). *"A Logical Calculus of the Ideas Immanent in Nervous Activity"*. *Bulletin of Mathematical Biophysics*.
- [McCulloch, 1988] McCULLOCH, W. S. (1988). *"Embodiments of Mind"*. Cambridge, MA: MIT Press.

- [Min&Papert, 1969] MINSKY, M. L., & S. A. PAPERT. (1988). *"Perceptrons". Expanded Edition. Cambridge, MA: MIT Press. Original edition, 1969.*
- [Moncrieff, 1961] MONCRIEFF, R. W. (1961). *"An instrument for measuring and classifying odours". J. Appl. Physiol., 16: 742.*
- [Oliveira, 2002] OLIVEIRA, Alessandra Guaracy. *"Implementação de um Sistema de Tratamento e Reconhecimento de Padrões para uma Microbalança de Quartzo utilizando Redes Neurais Artificiais". Dissertação final para obtenção do título de Bacharel em Ciência da Computação pela Universidade Federal de Lavras. Lavras / MG.2002.*
- [Parker, 1985] PARKER, D. *"Learning Logic". Technical Report TR-87, Cambridge, MA: Center for Computational Research in Economics and Management Science, MIT.*
- [Persaud e Dodd, 1982] PERSAUD, K. e DODD, G. H. (1982). *"Analysis of discrimination mechanisms of the mammalian olfactory system using a model nose". Nature, 299: 352-355.*
- [Pimenta, 1995] PIMENTA, Carlos José. *"Qualidade do café (Coffea arabica L.) originado de frutos colhidos em quatro estadios de maturação". 1995. 94 p. Dissertação (Mestrado em Ciência dos Alimentos) - Universidade Federal de Lavras, Lavras / MG.*
- [Rich, 1993] RICH, Elaine. *"Inteligência Artificial". São Paulo: Makron Books, 1993.*
- [Rosenblatt, 1958] ROSENBLATT, F. *"The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain". Psychological Review, 65:386-408. Reprinted in Anderson & Rosenfeld [1988], pp. 92-114.*
- [Wilk&Hat, 1964] WILKENS, W. F. e HATMAN, A. D. (1964). *"An electronic analog for the olfactory processes". Ann. NY Acad. Sci., 116: 608.*
- [Yamazaki, 2001] YAMAZAKI, Akio. *"Reconhecimento de padrões em um nariz artificial por redes neurais". 2001. 113 p. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Pernambuco, Recife / PE.*